CLAIMS

What is claimed is:

1. A method for enforcing ordering between read and write transactions for an adapter unit configured to connect to a bus, the adapter unit having at least one read and at least one write buffer, comprising:

enqueuing, in the write buffer, an initiator write transaction for performance on the bus;

receiving, in the read buffer, an initiator read transaction request for performance on the bus, the performance, on the bus, of the initiator read transaction request being required to follow the performance, on the bus, of the initiator write transaction; and

if, after receiving the initiator read transaction request, the performance of the initiator write transaction has not started on the bus,

performing the initiator write transaction on the bus; and

then performing the initiator read transaction request on the bus.

2. A method for enforcing ordering between read and write transactions, as recited in claim 1, further comprising:

enqueuing a target read completion transaction in the write buffer, the target read completion transaction having a position in the write buffer either before or after the initiator write transaction based on whether the target read completion originally occurred before or after the initiator write transaction; and

performing on the bus the transactions in the write buffer in the order in which the transactions were enqueued.

3. A method for enforcing ordering between read and write transactions, as recited in claim 1,

further comprising, if, after receiving the initiator read transaction request, the initiator write transaction has not started on the bus, the step of setting a flag associated with the initiator read transaction request indicating that the initiator write transaction has not started on the bus;

wherein the step of performing the initiator write transaction on the bus includes clearing the flag; and

wherein the step of performing the initiator read transaction request on the bus includes performing the initiator read transaction request on the bus if the flag is not set.

4. A method for enforcing ordering between read and write transactions for an adapter unit configured to connect to a bus, the adapter unit having at least one read and at least one write buffer, comprising:

enqueuing, in the write buffer, an initiator write transaction to be performed on the bus;

receiving, in the read buffer, an initiator read transaction request to be performed on the bus, the performance, on the bus, of the initiator read transaction request being required to follow the performance, on the bus, of the initiator write transaction;

clearing a write contingency flag in a register associated with the read transaction request;

if, after receiving the read transaction request, the performance of the initiator write transaction has not started on the bus:

setting the write contingency flag in the register to indicate the dependence of the initiator read transaction request on the initiator write transaction in the write buffer;

performing the initiator write transaction on the bus;

after the performance of the write transaction is completed, clearing the write contingency flag in the register; and

if the write contingency flag is not set, performing the read transaction request on the bus.

5. A method for enforcing ordering between read and write transactions, as recited in claim 4, further comprising:

enqueuing a target read completion transaction in the write buffer, the target read completion transaction having a position in the write buffer either before or after the initiator write transaction based on whether the target read completion originally occurred before or after the initiator write transaction; and

performing on the bus the transactions in the write buffer in the order in which the transactions were enqueued.

6. An adapter unit configured to connect to a bus, the adapter unit comprising:

a bus protocol logic unit connected to the bus, for performing read and write transactions on the bus according to a pre-established bus protocol;

buffer manager logic unit connected to the bus protocol logic unit, for enforcing ordering of performance of read and write transactions on the bus, the buffer manager logic unit including:

a plurality of read buffers connected to provide read transaction data to the bus protocol unit;

a plurality of write buffers connected to provide write transaction data to the bus protocol unit;

a plurality of write context registers, each write context register associated with one of the write buffers;

a plurality of read context registers, each read context register being associated with one of the read buffers and having a write contingency bit associated with each write buffer of the plurality of write buffers, the buffer manager logic unit being configured to:

initiate performance of a read transaction request or a write transaction on the bus, the performance of a read transaction request being dependent on whether a write contingency flag is set in the read context register associated with the read transaction request;

enqueue, in the write buffer, an initiator write transaction to be performed on the bus;

receive, in the read buffer, an initiator read transaction request to be performed on the bus;

set a write contingency flag in the read context register associated with a received read transaction request, when performance, on the bus, of a received initiator read transaction request is required to follow the performance, on the bus, of an enqueued initiator write transaction; and

clear a write contingency flag in the read context register, when performance, on the bus, of an enqueued initiator write transaction is completed; and

an application specific logic unit connected to the buffer manager logic unit, and configured to perform functions specific to the adapter unit by providing initiator read request transactions and write transactions to the buffer manager logic unit.

5